Abstract

An apparatus and method for universal programming language conversion between two different sequential programming languages, e.g., sequential procedural and sequential object oriented programming languages. In particular, conversion is between a source program in a first programming language and a target program in a second programming language. Initially, the source program in the first programming language is parsed using a parsing interface specific to the first programming language. All syntax from the parsed source program is then stripped or removed. Classes in a framework are instantiated to capture semantics of the parsed source program independent of syntax and execution model of the sequential programming languages. The classes are C++ classes representing fundamental core constructs of all sequential programming languages. A semantic representation of the parsed source program without any syntax is produced. The semantic representation is received at a printer interface specific to the second programming language and syntax of the target program in the second programming language is added. This same process can be used for either high-level conversion or compilation depending on whether the target programming language is high level or low level, respectively.